The University of Alabama in Huntsville
ECE Department
CPE 526 01
Midterm Exam
February 26, 2004

1. (20 points) Describe the following logic expression

(A'• B' • C • D) + (A • B • C) + (B' • C')

with a structural VHDL model using the following package located in library WORK.

Hint: If you don't need all of the inputs, you can tie one or more of them to '0' or '1'.

```
package LOGIC_PKG is
  component AND2_OP
    port A, B, C : in BIT; Z out BIT);
  end component;
  component NAND2_OP
    port A, B : in BIT; Z out BIT);
  end component;
  component OR4_OP
    port A, B, C, D : in BIT; Z out BIT);
  end component;
end LOGIC_PKG;
```

2. (1 point) A ______________________________________ is a high-level programming language with specialized constructs for modeling hardware.

3. (15 points). Write a VHDL function that accepts a bit vector of arbitrary length and returns a bit vector that is the reverse of the input. For example:

Input:	0101111
Output:	1111010

4. (1 point) ______________ delay is the delay which represents gate delay in VHDL.

5. (1 point) All statements inside of a block are ______________________.

6. (3 points) For the following function call, which function will be called? _____

```
VARIABLE a, b : INTEGER;
b := decrement (a);
```

```
(a)   FUNCTION decrement (x : INTEGER) RETURN INTEGER;
(b)   FUNCTION decrement (x : REAL) RETURN REAL;
```

7. (1 point) A(n) ___________________ is an aggregate type in VHDL.

8. (15 points) Design a hardware device that can translate from Gray code to BCD code. The device inputs will be the four bits of the Gray code, and the device outputs will be the four bits of the equivalent BCD code. The illegal input combinations may be treated as don't care conditions.

Decimal Digit	Gray Code	BCD Code
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	1110	0101
6	1010	0110
7	1011	0111
8	1001	1000
9	1000	1001

(a) (4 points) Write an entity for the device. (b) (11 points) Use concurrent signal assignments to model the device.

9. (6 points) Consider the following structural VHDL model.

```
entity SMODEL is
   port
     (P1 : in BIT;
      P2 : out BIT;
      P3 : inout BIT);
end SMODEL;

architecture STRUCTURE of SMODEL is
   component UNIT
      port (C1, C2, : in BIT; C3 : out BIT);
   end component;

begin
  U1 : UNIT port map (C1 => ?, C2 => ?, C3 => ?);
end STRUCTURE;
```

(a) (3 points) Complete the structural description by giving a legal set of port-to-port connections for entity ports P1, P2, and P3 and component ports C1, C2, and C3.

(b) (3 points) Is there more than one possible set of legal port-to-port connections?

10. (22 points) Specify type declarations for the following data types.

a. (4 points) A three valued logic system, `MVL3`, with values ‘0’, ‘1’, and ‘Z’. Values ‘0’ and ‘1’ have the usual logic meaning and ‘X’ means unknown. Any uninitialized data item of this type should have value ‘Z’.

b. (4 points) A `SEASON_OF_YEAR` enumeration data type.

c. (2 points) A data type `MONTH_NUMBER` that can have integer values in the range from 1 to 12.

d. (2 points) A data type `COST` that can have real values between $0.00 and $1,405.00.

e. (2 points) A descending range data type `HIGH_WORD` with integer values from 63 to 32.

f. (4 points) A 32-bit descending-index register composite data type, `REG_32_HIGH`, with index valued from the type `HIGH_WORD` declared above, and component values of type MVL3.

g. (4 points) A three-dimensional table, `TABLE_3D`, with index values and table entries all of type `std_logic` (which has been declared elsewhere and is visible).

11. (1 point) An entity X, when used in another entity, becomes a __________________ for the entity Y.

12. (1 point).__________________ is an example of a built-in enumerated type.

13. (12 points) Draw the state diagram for the following state machine. Is it a Moore machine or a Mealy machine?

```
ENTITY state_machine IS
   PORT (sig_in ; IN BIT; clk : IN BIT;
          sig_out : OUT BIT);
END state_machine;

ARCHITECTURE state_machine OF state_machine IS
   TYPE state_type IS (a, b, c, d, e);
   SIGNAL current_state, next_state : state_type;
BEGIN
   PROCESS (sig_in, current_state)
   BEGIN
      sig_out <= ‘0’ ;
      next_state <= b;
      CASE current_state
      WHEN a =>
         IF sig_in = ‘0’  THEN
            next_state <= a;
         ELSE
            next_state <= d;
         END IF;
         sig_out <= ‘1’ ;
      WHEN b =>
         IF sig_in = ‘0’  THEN
            next_state <= b;
         ELSE
            next_state <= c;
         END IF;
     WHEN c =>
       IF sig_in = ‘1’  THEN
           next_state <= a;
       ELSE
          next_state <= d;
       END IF;
       sig_out <= ‘1’ ;
     WHEN d =>
       IF sig_in = ‘0’  THEN
           next_state <= e;
       END IF;
     WHEN e =>
       IF sig_in = ‘1’  THEN
          next_state <= c;
       END IF;
      END CASE;
   END PROCESS;
   PROCESS (clk)
   BEGIN
      IF (clk’ EVENT AND clk = ‘1’ ) THEN
          current_state <= next_state;
      END IF;
   END PROCESS;
END state_machine;
```

14. ____________________________ is the hardest problem.